RECONCILIATION WITH NON-BINARY SPECIES TREES

B. Vernot, M. Stolzer, A. Goldman, D. Durand*

Departments of Biological Sciences and Computer Science, Carnegie Mellon University,
Pittsburgh, Pennsylvania 15213, USA
Email: {bvernot,mstolzer,aiton,durand}@{cs,andrew,cs,cs}cmu.edu

Reconciliation is the process of resolving disagreement between gene and species trees, by invoking gene duplications and losses to explain topological incongruence. The resulting inferred duplication histories are a valuable source of information for a broad range of biological applications, including ortholog identification, estimating gene duplication times, and rooting and correcting gene trees. Reconciliation for binary trees is a tractable and well studied problem. However, a striking proportion of species trees are non-binary. For example, 64% of branch points in the NCBI taxonomy have three or more children. When applied to non-binary species trees, current algorithms overestimate the number of duplications because they cannot distinguish between duplication and deep coalescence. We present the first formal algorithm for reconciling binary gene trees with non-binary species trees under a duplication-loss parsimony model. Using a space efficient mapping from gene to species tree, our algorithm infers the minimum number of duplications and losses in $O(|V_G| \cdot (k_S + h_S))$ time, where V_G is the number of nodes in the gene tree, h_S is the height of the species tree and k_S is the width of its largest multifurcation. We also present a dynamic programming algorithm for a combined loss model, in which losses in sibling species may be represented as a single loss in the common ancestor. Our algorithms have been implemented in NOTUNG, a robust, production quality tree-fitting program, which provides a graphical user interface for exploratory analysis and also supports automated, high-throughput analysis of large data sets.

Keywords: Reconciliation, non-binary species trees, polytomy, gene duplication, gene loss, lineage sorting, deep coalescence

.

1. INTRODUCTION

Reconciliation is the process of constructing a mapping between a gene family tree and a species tree. Under a model of duplication-loss parsimony, this mapping can be used to infer duplications and losses in the history of the gene family, as well as the species lineages in which these events occurred.

Reconciliation is an essential method in molecular phylogenetics that is widely used in evolutionary applications in medicine, development biology and plant science. Reconciliation, following phylogeny reconstruction, is the most reliable approach for identifying orthologs for use in function prediction, gene annotation, planning experiments in model organisms, and identifying drug targets (e.g., Refs. 5, 40). Reconciliation is used to correlate specific duplications with the emergence of novel cellular functions or morphological features, providing clues to the functions of newly discovered genes (e.g., Refs. 8, 47). Minimizing duplications and losses provides a basis for rooting an unrooted tree[7] and for selecting alternate gene or species tree topologies.[5,7,10,26] Reconciliation is also the kernel of a related, but more complex, problem: inferring a species tree from many gene trees.[21] High throughput reconciliation tools are required for automated construction of databases of molecular phylogenies.[9,19,33,37]

Reconciliation of binary trees is a well-studied problem and a number of software packages for this problem are available.[9,10,28,29,37,49,50] However, standard reconciliation will not produce correct results when applied to a non-binary species tree. Discordance between a binary gene tree and a binary species tree is always evidence of gene duplication. In contrast, when the species tree is non-binary, two different processes can cause discordance between gene and species trees: gene duplication and incomplete lineage sorting. Since these are different biological phenomena with different consequences for the interpretation of phylogenetic studies, it is essential to distinguish between discordances that *must* be due to duplication (*required* duplications) and discordances that could be due to either duplication or incomplete lineage sorting (*conditional* duplications). Standard binary reconciliation cannot make this distinction.

As the tree of life project gains momentum, it is

[*] Corresponding Author. durand@cs.cmu.edu

becoming evident that this is not a minor problem relegated to a few obscure species lineages. Rather, sixty four percent of branch points in the NCBI taxonomy,[46] one of the most widely used databases of species phylogenies, have more than two children. In addition, a number of well documented analyses of simultaneous divergences have been reported.[15, 18, 24, 34, 39] Reconciliation methods for non-binary trees are urgently needed. To our knowledge, no formal algorithms for reconciliation of non-binary species trees have been published.

Our contributions: To address this need, we present novel algorithms to find the minimum number of duplications and losses when reconciling a binary gene tree, $T_G = (V_G, E_G)$, with a non-binary species tree, $T_S = (V_S, E_S)$. We construct a mapping from nodes in T_G to *sets of nodes* in T_S that allows us to test efficiently whether a discordance at a given node is a conditional or required duplication. Our mapping is space efficient; the maximum size of the set labeling any node in T_G is $O(k_S)$, where k_S is the maximum outdegree in T_S.

Using this mapping, we present an efficient algorithm for reconciling a binary gene tree with a non-binary species tree under duplication-loss parsimony. Our algorithm infers all conditional and required duplications in $O(|V_G| \cdot (k_S + h_S))$ time, where h_S is the height of T_S.

We also present algorithms to infer the minimum number of gene losses. We estimate the time at which each loss occurred by assigning it to an edge in T_G. For binary species trees, this assignment is unambiguously determined by the reconciliation and is easily calculated. However, the lack of resolution in a non-binary tree leads to uncertainty about exactly when a loss occurred. Parsimony provides a principled basis for reducing this uncertainty: we assign losses in such a way that the total number of losses is minimized.

We propose two minimization criteria and provide algorithms to infer losses for each one. The first criterion is based on the assumption that all losses were independent events (*explicit losses*). We infer the minimum set of explicit losses, again in $O(|V_G| \cdot (k_S + h_S))$ time. The second criterion is motivated by the observation that, under certain circumstances, losses in sibling species can be explained by a single loss in their common ancestor. Under this assumption, we can reduce the number of losses by combining losses that share a parent, whenever possible. We present a dynamic program that minimizes the number of *combined losses*, by considering all combinations of possible edge assignments for each loss to maximize the opportunities to combine losses. The worst case time complexity of this algorithm is exponential, but its performance is good on real trees from typical data sets, as we demonstrate empirically.

Our algorithms have been implemented in NOTUNG, a software package that takes trees in the widely used Newick format as input, permitting interoperability with a wide range of phylogeny reconstruction packages. The resulting reconciliation can be viewed and manipulated using an interactive graphical user interface. A batch processing interface for automated analysis of large phylogenetic data sets is also available. Our software is implemented in Java and runs on Windows, Unix and Mac OS X. It is freely available at `http://www.cs.cmu.edu/~durand/Notung`.

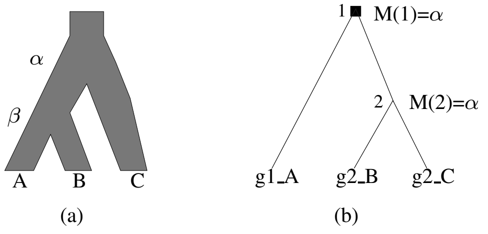
Roadmap: In the next section, we introduce notation, review the standard algorithm for reconciliation of binary trees, and summarize previous work on binary reconciliation. In Section 3, we review the relevant models of non-binary gene and species trees in the molecular evolution literature and give formal definitions for required and conditional duplications based on this foundation. Next we present our non-binary reconciliation algorithms. Duplications are discussed in Section 4. In Section 5, we present algorithms for inferring the minimum number of explicit and combined losses and compare our work to related work by other authors. In Section 6, we demonstrate the utility of our methods with analyses of real data sets using our software. In the conclusion, we discuss probabilistic approaches to reconciliation and describe directions for future work.

2. NOTATION AND BINARY RECONCILIATION

In this section, we introduce notation and review the standard reconciliation algorithm for binary trees.

Let $T_i = (V_i, E_i)$ be a rooted tree, where V_i is the set of nodes in T_i, and E_i is the set of edges. $L(T_i)$ is the leaf set of T_i and $L(v)$ refers to the leaf set of a subtree rooted at $v \in V_i$. $C(v)$ and $p(v)$ refer to the children and parent of v, respectively. If v is binary, $r(v)$ and $l(v)$ denote the right and left children of v. A non-binary node in a tree is referred to as an unresolved node or *polytomy*. A group of taxa is *monophyletic* if it corresponds to an ancestor and all of its descendants. The root node of T_i is $root(T_i)$. If $u \in V_i$ lies on the path from v to $root(T_i)$, we say that $u \geq_i v$. We follow the computer science convention, in which the root is at the top of the tree, the leaves are at the bottom, and $p(g)$ is above g. In tree figures, g_s denotes a gene that is sampled from species s.

The objective of reconciliation is to identify gene duplications and losses by fitting a gene tree to a species tree.[12,30,36] Let T_G be a binary gene tree and T_S be a binary species tree such that the genes in $L(T_G)$ were sampled from the species in $L(T_S)$. A mapping $M(\cdot)$ is constructed from each node $g \in V_G$ to a target node $s \in V_S$. If $g \in L(T_G)$, $M(g)$ is the species from which g was sampled. Otherwise, $M(g)$ is the least common ancestor (*LCA*) of the target nodes of its children, *i.e.* $M(g) = LCA(M(l(g)), M(r(g)))$. Using this mapping, $g \in T_G$ is a *duplication* if the children of g map to the same lineage as g; *i.e.* if $M(g) = M(l(g))$ and/or $M(g) = M(r(g))$. Otherwise, g is a *speciation*. A duplication at g indicates that a duplication occurred within the lineage leading to the ancestral species, $M(g)$, and that two copies of the gene must have been present in $M(g)$. In Fig. 1, for example, although the gene tree contains one gene sampled from each species, the topological disagreement between gene and species trees implies a duplication. Both nodes 1 and 2 in the gene tree map to α. Since $M(1) = M(2)$, a duplication at node 1 is inferred. Losses can also be reconstructed from the mapping, $M(\cdot)$, as described in Ref. 10. We refer to this algorithm as *LCA reconciliation* in order to distinguish it from the new reconciliation algorithms proposed for non-binary species trees in the next section.



(a)

(b)

Fig. 1. Least Common Ancestor reconciliation. **(a)** Binary species tree. **(b)** Binary gene tree, reconciled with species tree (a). The black square indicates a duplication.

Variants of LCA reconciliation have been proposed by numerous authors, and several software packages for analyzing gene duplication histories have been developed.[9,10,28,29,37,49,50] A related problem, inferring the optimal species tree from multiple conflicting gene trees, has been studied extensively[13,14,21,25,27,42,48] for various optimization criteria[11] and has been shown to be NP-hard.[21]

3. MODELS FOR NON-BINARY SPECIES TREES

Least Common Ancestor reconciliation is based on the assumption that disagreement between a gene tree and a species tree indicates that one or more gene duplications must have occurred. In this section, we show that when the species tree is non-binary, this assumption is no longer warranted.

A polytomy may represent the simultaneous divergence of all descendants (a *hard polytomy*[22]). It may also indicate that the true binary branching process cannot be reconstructed (a *soft polytomy*[22]); this often occurs when a sequence of binary divisions proceeds in close succession and the time between these events is insufficient to accumulate informative variation.

Since the true branching pattern in a gene tree is always binary,[16] a polytomy in T_G can only represent uncertainty. In contrast, since a species tree represents the evolution of a population of organisms, a polytomy in the species tree may represent either simultaneous divergence or uncertainty. Simultaneous divergences of three or more lineages may result from the isolation of subpopulations within a widespread species by sudden meteorological or geological events, or from rapid expansion of the population into open territory, resulting in reproductive isolation. Examples of simultaneous divergences in nature include *Anolis* lizards,[18] modern birds in the order *Neoaves*,[34] macaque monkeys,[15,24] auklets,[45] and African cichlid fishes.[39]

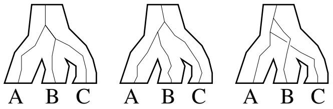


Fig. 2. Gene trees evolving within the same species polytomy can have different binary topologies.

Because the species tree represents a population with genetic diversity, gene trees with different binary branching processes can be consistent with a species polytomy.[20,23] If k or more alleles are present in the population when k lineages separate, a different allele may fix in each lineage. The resulting gene tree will be binary and will reflect the order in which new alleles arose in the ancestral population. This process is called *incomplete lineage sorting*. When the time of separation of lineages in T_G predates the time of speciation, the divergence is called a *deep coalescence*. As shown in Fig. 2, all possible binary gene trees with k leaves can occur in a k-tomy in the species tree. Deep coalescence can also occur when two or more binary speciation events occur in rapid succession. If the time between subsequent speciations is shorter than the fixation time, more than one allele will still be present at the time of the second speciation (see, for example, Ref. 35). The probability of this occurring increases with shorter branch lengths and larger effective population sizes.[16,23,31,43,44]

When there is simultaneous or rapid divergence in the species tree, the challenge is to determine whether disagreement between a gene and specie tree indicates a deep coalescence or a duplication. Some incongruences can only be explained by a duplication. Obviously, a duplication must have occurred in any gene family that has two or more members in the same species. Even when no contemporary species contains more than one family member, there are cases where topological disagreement can only be explained by a duplication. For example, the incongruence between the species tree in Fig. 3(a) and the gene tree in Fig. 3(b) can only be explained by a duplication at node 2. This can be seen in Fig. 3(c), which shows the gene tree embedded in the species tree. Two copies of the gene are present in the ancestral species β, indicating that a duplication must have occurred. We refer to cases where a duplication must have occurred as *required duplications*. In other cases, it is not possible to determine whether the disagreement is due to a deep coalescence or duplication.[41] For example, node 1 in Fig. 3(c) is associated with a deep coalescence rather than a duplication; however, it is possible that this node could also have been a duplication. These instances will be referred to as *conditional duplications*.

Next, we formally characterize the properties of gene and species trees that determine when a duplication is required. For each polytomy $s \in V_S$, let $H(s)$ be the set of all possible binary subtrees whose leaves are the children of s. Formally, given the k-tomy $s \in V_S$, let $H(s) = \{T_i | L(T_i) = C(s)\}$. For example if node s is the trichotomy α in Fig. 3(a), then $H(s) = \{(A,(B,\beta)),(B,(A,\beta)),(\beta,(A,B))\}$. Let $H^*(T_S)$ be the set of all possible binary trees obtained by replacing each polytomy $s_i \in V_S$ with each tree $T_{ij} \in H(s_i)$. Note that, for every $T' \in H^*(T_S)$, every node $s \in T_S$ corresponds to a node in T'; however, T' will also contain nodes that do not correspond to any node in T_S. The cardinality of $H^*(T_S)$ is $\prod_{\forall s_i \in T_S} |H(s_i)|$, which is equivalent to $\prod_i n_i$, where $n_i = \frac{(2k_i-3)!}{2^{k_i-2}(k_i-2)!}$ and $k_i = |C(s_i)|$. If T_S is binary, then $H^*(s) = \{T_S\}$.

We now use $H^*(T_S)$ to characterize formally the properties of the gene and species tree that determine when a duplication is required. When reconciling T_G with every $T' \in H^*(s)$, if $g \in V_G$ is a duplication in every reconciliation, then a duplication *must* have occurred. If at least one, but not all reconciliations indicate a duplication at g, then a deep coalescence *may* have occurred. Formally:

Definition 3.1. $\forall T' \in H^*(T_S)$, reconcile T_G with T'. Given $g \in V_G \setminus root(T_G)$

$p(g)$ is a *required duplication* if $\forall T' \in H^*(T_S)$, $M(g) = M(p(g))$.

$p(g)$ is a *conditional duplication* if $\exists T' \in H^*(T_S)$ s.t. $M(g) = M(p(g))$ and $p(g)$ is not a required duplication.

In the example in Fig. 3, $H^*(T_S) = \{(A,(B,(C,D))),(B,(A,(C,D))),((A,B),(C,D))\}$. From Definition 3.1, node 2 is a required duplication, since for every $T' \in H^*(T_S)$, $M(2) = M(3)$. Node 1 is a conditional duplication since there is a binary resolution in $H^*(T_S)$ — $(A,(B,(C,D)))$ — that does not lead to disagreement at 1.

4. IDENTIFYING DUPLICATIONS

The goal of reconciliation with non-binary species trees is to determine whether a given node is a required duplication, a conditional duplication, or a speciation. Definition 3.1 provides a formal basis for such a test, but cannot be the basis of an efficient algorithm, since $H^*(T_S)$ grows superexponentially with the size and number of polytomies in T_S.

While *LCA* reconciliation is able to identify all conditional and required duplications, it cannot distinguish between the two. For example, in the gene tree in Fig. 3(b), nodes 1, 2, and 3 are all mapped to α, indicating a duplication at nodes 1 and 2 under Least Common

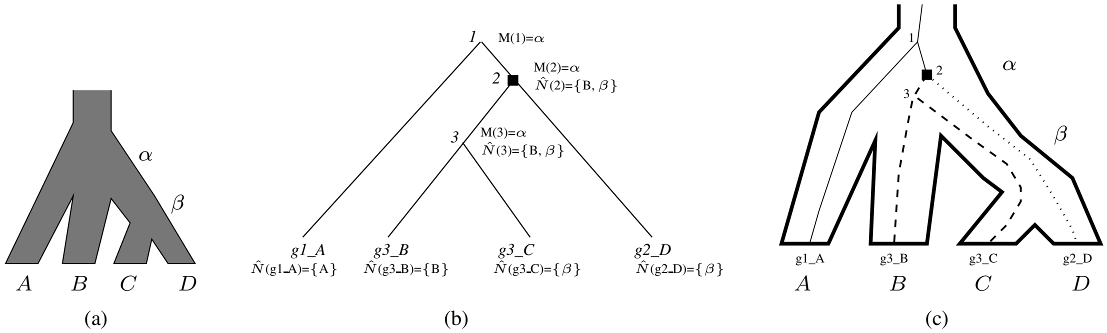


Fig. 3. **(a)** A species tree with a polytomy at α. **(b)** A gene tree reconciled with species tree (a). **(c)** The gene tree (b) embedded in species tree (a). Black squares indicate duplications. Losses are not represented in these trees.

Ancestor reconciliation. While this inference is correct for node 2, it incorrectly infers a duplication at 1. Why is Least Common Ancestor reconciliation unable to distinguish between required and conditional duplications? $M(g) = s$ implies that the ancestral gene g was present in the ancestral population, s. If T_S is binary, then the descendants of g were also present in all descendants[a] of s. This, in turn, implies that when $M(p(g)) = M(g)$, both g and its parent were present in the same species, s, indicating that a duplication occurred at $p(g)$. This reasoning is the basis of Least Common Ancestor reconciliation.

However, when T_S is non-binary, it is not necessarily true that the descendants of g were present in all descendants of $M(g)$. For example, in Fig. 3, $M(2) = \alpha$, but no descendant of 2 is present in species A, which is a descendant of α. This is due to the deep coalescence at node 1 in T_G. As this example shows, $M(\cdot)$ does not contain the information required to infer the set of nodes in T_S in which the descendants of g must have been present. In order to distinguish between conditional and required duplications, we need a new mapping from T_G to T_S that allows us to determine when more than one descendant of g was present in some descendant of $M(g)$.

We propose a mapping in which nodes in T_G are mapped to sets of nodes in T_S. A naïve solution would be to decorate each node in T_G with all of the nodes in T_S in which the descendants of g were present; *i.e.*, with all nodes in the subtree rooted at $M(g)$. However, this creates a problem of efficiency: The size of the sets labeling the nodes in the gene tree grows with the height of the tree and can contain as many as $O(|V_S|)$ elements. Fortunately, it is sufficient to store the roots of the subtrees in which descendants of g must have been present. The mapping presented in Definition 4.1 is sufficiently informative to identify required and conditional duplications. Moreover, the size of this mapping at any given node is bounded by the size of the largest polytomy in T_S.

Definition 4.1. Define $\hat{N} : V_G \setminus root(T_G) \rightarrow V_S^+$ to be $\hat{N}(g) = \{M(g)\}$ if $M(p(g)) \in L(T_S)$. Otherwise, $\hat{N}(g) =$
$\{h | h \in C(M(p(g))) \wedge \exists\, v \in L(g) \ni h \geq_S M(v)\}$

For a given gene, g, in species $s = M(g)$, $\hat{N}(g)$ is the set of species that are children of the parent of s in which descendants of g must be present. For example, in Fig. 3(b), $\hat{N}(3) = \{B, \beta\}$ because $M(p(3)) = M(2) = \alpha$, and descendants of 3 were present in B and β, which are in $C(\alpha)$. Note that $\hat{N}$ is defined on every node in T_G except the root. $\hat{N}$ provides an efficient and accurate test for required duplications:

Theorem 4.1. *A node g is a required duplication iff $\hat{N}(r(g)) \cap \hat{N}(l(g)) \neq \emptyset$.*

Proof.
$\rightarrow$ If $\hat{N}(l(g))$ and $\hat{N}(r(g))$ intersect, then they share at least one element, x. Thus, for all $T' \in H^*(T_S)$, both $M(l(g))$ and $M(r(g))$ must be x or ancestors of x, and thus both lie on the path from x to $M(g)$. This requires that either $M(l(g)) = M(r(g))$ or one is a descendant of the other, and g meets the duplication criterion for binary gene and species trees.
$\leftarrow$ We need to show that whenever $\hat{N}(r(g)) \cap \hat{N}(l(g)) = \emptyset$, there is always at least one element of $H^*(T_S)$ that does not imply a duplication at g. Any $T' \in H^*(T_S)$ that has all members of $\hat{N}(l(g))$ in the left subtree of $M(g)$

[a] unless g was lost. However, in that case we would need to infer a loss in a descendant of s.

and all members of $\hat{N}(r(g))$ in the right subtree of $M(g)$ will meet this criterion. □

Corollary 4.1. *Node $g \in V_G$ is a conditional duplication if g is not a required duplication and if $M(g) = M(l(g))$ and/or $M(g) = M(r(g))$.*

Using Definition 4.1 we can correctly classify nodes in Fig. 3(b). Since $\hat{N}(3) = \{B, \beta\}$ and $\hat{N}(g2\_D) = \{\beta\}$, their intersection is $\{\beta\}$, indicating the presence of two genes in ancestral taxa β. Therefore, a duplication is inferred at node 2. However, $\hat{N}(g1\_A) = \{A\}$ and $\hat{N}(2) = \{B, \beta\}$. $\hat{N}(1) \cap \hat{N}(2) = \emptyset$, correctly implying that no duplication is required at gene node 1.

$\hat{N}(g)$ is calculated easily with a postorder traversal of T_G and is a function of the union of $\hat{N}(l(g))$ and $\hat{N}(r(g))$. An additional *climbing* step must be taken to ensure that the set $\hat{N}(g)$ is composed only of children of $M(p(g))$. The *climb* procedure, given in Alg. 5.1, also prevents $|\hat{N}(\cdot)|$ from growing larger than k_S. Alg. 5.1 infers a required duplication at g if the intersection of the sets $\hat{N}(l(g))$ and $\hat{N}(r(g))$ is non-empty. Conditional duplications are inferred using Least Common Ancestor reconciliation. $\hat{N}(\cdot)$ can also be used to infer duplications when the species tree is binary, and produces the same results as Least Common Ancestor reconciliation. Pseudocode for this algorithm, which also calculates explicit losses, is given in Alg. 5.1 in the next section.

Theorem 4.2. ***Equivalence with LCA Reconciliation for Binary Species Trees.*** *Let T_G be a binary gene tree reconciled with a binary species tree. $\hat{N}(r(g)) \cap \hat{N}(l(g)) \neq \emptyset$ iff $M(r(g)) = M(g)$ and/or $M(l(g)) = M(g)$*

Proof.

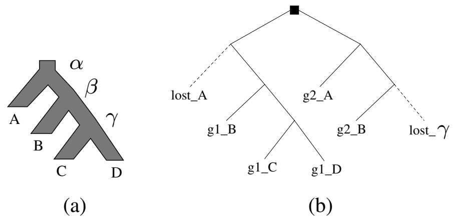
$\rightarrow$ This follows directly from Theorem 4.1.

$\leftarrow$ Suppose that there exists $h \in C(g)$, such that $M(h) = M(g)$, but $\hat{N}(r(g)) \cap \hat{N}(l(g)) = \emptyset$. There are two cases: either $\hat{N}(l(g))$ and $\hat{N}(r(g))$ are both equal to $\{M(g)\}$ or $\hat{N}(l(g))$ and $\hat{N}(r(g))$ both contain children of $M(g)$. In the former case, $\hat{N}(l(g))$ and $\hat{N}(r(g))$ are not disjoint, leading to a contradiction. In the latter case, $\hat{N}(l(g))$ and $\hat{N}(r(g))$ both contain children of $M(g)$. Since $M(g)$ has only two children and the sets are disjoint, one set must contain the right child of $M(g)$ and the other the left child. Then, $M(r(g))$ and $M(l(g))$ are not in the same lineage and there is no child $h \in C(g)$, such that $M(h) = M(g)$, leading to a contradiction. □

5. IDENTIFYING LOSS NODES

In this section, we discuss how to infer gene losses when reconciling with non-binary species trees. For a given gene tree, T_G, and species tree, T_S, we report the total number of losses in T_G and the timing of individual losses. We designate the time period when a loss occurred by assigning it to an edge in T_G.

When binary gene trees are reconciled with binary species trees under a parsimony model, the placement of each loss is unambiguous, and can be determined efficiently from $M(\cdot)$.[7] For example, in the gene tree in Fig. 4(b), three losses have occurred in the contemporary species A, C and D. Note that the two losses in C and D can be explained by the loss of a single ancestral gene in the ancestral species γ. In general, given a set of monophyletic losses in a reconciled gene tree, it is more parsimonious to infer a single loss at the root of the corresponding clade in T_S.



(a) (b)

Fig. 4. Losses in Least Common Ancestor reconciliation. **(a)** A binary species tree. **(b)** A binary gene tree reconciled with species tree (a).

In contrast to the binary case, when the species tree is non-binary, it is not always possible to determine exactly where a loss occurred. In Fig. 5(b), a descendant of node 2 is present in A and D but absent from B and C. This suggests losses in the lineages leading to the contemporary B and C species. However, for each loss it is not possible to determine whether the lost gene diverged before or after the divergence at node 2 and, if the latter, whether it was more closely related to $g2\_A$ or $g2\_D$. For example, for the gene lost in B, this leads to three possible loss scenarios: $(g2\_A, (lost\_B, g2\_D))$, $((g2\_A, lost\_B), g2\_D)$, or $(lost\_B, (g2\_A, g2\_D))$. Note that each placement of $lost\_B$ in this tree implies a different element of $H(\alpha)$.

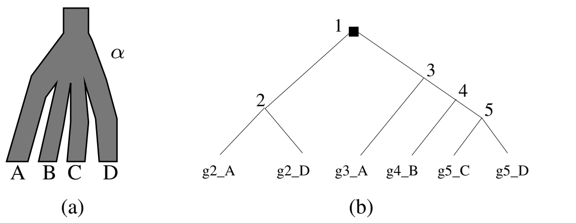


Fig. 5. **(a)** A non-binary species tree. **(b)** A gene tree reconciled with species tree (a). Losses are not shown.

For non-binary reconciliation, we combine any set of losses that potentially form a monophyletic subtree; *i.e.*, if a set of losses located on the same edge in T_G corresponds to a monophyletic subtree of some $T' \in H^*(T_S)$, they can be replaced with the single loss of an ancestral gene. A simple test is that if several lost genes (or roots of lost subtrees) on the same edge all map to siblings of the same polytomy, they can be combined.

Consider the losses in B and C in Fig. 5. If $lost\_B$ and $lost\_C$ are placed on the same edge, we can infer a single loss in their putative common ancestor. For example, if both are placed on the edge between 1 and 2, then they correspond to the monophyletic subtree $((A, D), (B, C)) \in H^*(T_S)$. Thus, the losses in B and C can be combined, yielding a single loss for this reconciliation. If the losses are placed on different edges, they cannot be combined; two losses will be inferred.

In the parsimony context, our goal is to assign loss nodes to edges in T_G so as to minimize the total number of inferred losses. Since only losses on the same edge can be combined, the choice of placement influences the total number of losses. With this in mind, we propose two minimization strategies. The first strategy assigns losses to edges so as to minimize the total number of uncombined (or explicit) losses, and then combines losses wherever possible. The second strategy considers all possible assignments of losses to edges in T_G and selects the assignment that minimizes the number of combined losses. Although each minimization strategy restricts the number of possible placements, it is still possible to have more than one optimal placement of losses. We present algorithms for both minimization strategies, described below.

5.1. Minimizing Explicit Losses

Minimization of explicit losses is straightforward because each loss can be assigned to an edge in T_G independently. In addition to $\hat{N}(\cdot)$, we introduce a second mapping $N(\cdot)$ that allows us to infer losses by taking the difference between $N(\cdot)$ and $\hat{N}(\cdot)$.

Definition 5.1. Define $N : V_G \to V_S^+$ to be $N(g) = \{M(g)\}$ if $M(g) \in L(T_S)$. Otherwise, $N(g) = \{h | (h \in C(M(g)) \wedge \exists\, v \in L(g) \ni h \geq_S M(v))\}$

$N(g)$ is the set of children of $M(g)$ such that the descendants of g were present in the descendants of each element in $N(g)$. Just as $\hat{N}(g)$ is a subset of the children of $M(p(g))$, $N(g)$ is a subset of the children of $M(g)$. Unlike $\hat{N}(g)$, $N(g)$ is defined for $root(T_G)$.

For a given edge $e = (g, p(g))$, we use $N(p(g))$ and $\hat{N}(g)$ to infer explicit losses on e. As described in Alg. 5.1, we make a single postorder traversal of T_G, in which we calculate $N(\cdot)$ and $\hat{N}(\cdot)$, and infer the explicit losses associated with each edge by applying the following four tests. Each test corresponds to one of the four situations that can incur a loss on e.

(1) **Binary Duplication Losses:** A duplication node and its children should be mapped to the same node in the species tree; otherwise, a loss has occurred. Formally, if $p(g)$ is a required duplication and $M(p(g))$ is binary, then if $M(p(g)) \neq M(g)$, the species in $N(p(g)) \setminus \hat{N}(g)$ are lost at e (lines 18-21 in Alg. 5.1).

(2) **Skipped Species Losses:** If a node and its parent do not correspond to child and parent nodes in the species tree, the intervening species must have been lost. If $M(g) \neq M(p(g))$ and $p(M(g)) \neq M(p(g))$, we climb in T_S from $M(g)$ to $M(p(g))$ and infer a loss at every skipped species (lines 29-31 in Alg. 5.1).

(3) **Polytomy Duplication Losses:** This is analogous to test (1). If $M(p(g))$ is a polytomy and $p(g)$ is a required duplication, then the species in $N(p(g)) \setminus \hat{N}(g)$ are lost at e (lines 18-21 in Alg. 5.1).

(4) **Polytomy Speciation Losses:** If a node g maps to a polytomy and maps to a different species than its parent, then all children of $M(g)$ should contain a descendant of g. Otherwise, a loss has occurred. Formally, if $M(p(g)) \neq M(g)$ and $M(g)$ is a polytomy, the species in $C(M(g)) \setminus N(g)$ are lost at e (lines 22-24 in Alg. 5.1).

We demonstrate these four tests on the right subtree of Fig. 6(c), which has been labeled with the minimum number of explicit losses. (We do not discuss losses in the left subtree.) The following losses occur in this subtree: On the edge between nodes 1 and 3, a Binary Duplication

Loss in $N(1) \setminus \hat{N}(3) = A$ and a Polytomy Speciation Loss in $C(\beta) \setminus N(3) = B$. On the edge between 3 and 4, a Polytomy Duplication Loss in $N(3) \setminus \hat{N}(4) = D$. On the edge between nodes 4 and $g4\_E$, a Skipped Species Loss in F. On the edge between nodes 3 and 5, a Polytomy Duplication Loss in C. Finally, on the edge between 5 and $g5\_F$, a Skipped Species Loss in E.

The rules described above minimize explicit losses by assigning each loss as close to $root(T_G)$ as possible. Any other placement might move the loss below a duplication, which would require two losses, one in each subtree of the duplication node. Once explicit losses are identified, those losses which form a clade in some $T' \in H^*(T_S)$ and are placed on the same edge in T_G can be combined.

Algorithm 5.1.

```
reconcile( g )
  if ( g.isLeaf() )
    $M(g) = $ species of g
    $N(g) = \{M(g)\}$
    return
  // INTERNAL NODE CASE
  // descend first
  reconcile($l(g)$); reconcile($r(g)$)
  $M(g) = LCA(M(l(g)), M(r(g)))$
  calculateRequiredDuplication( g )
  if ( g $\neq$ Required Duplication )
    if ( $M(g) == M(l(g)) \,||\, M(g) == M(r(g))$ )
      g is Conditional Duplication

calculateRequiredDuplication( g )
  $\hat{N}(l(g)) = climb(l(g), g)$
  $\hat{N}(r(g)) = climb(r(g), g)$
  $N(g) = \hat{N}(l(g)) \cup \hat{N}(r(g))$
  if ( $\hat{N}(l(g)) \cap \hat{N}(r(g)) \neq \emptyset$ )
    g is Required Duplication
    // duplication losses for left child
    Losses($l(g)$) += $N(g) \setminus \hat{N}(l(g))$
    // duplication losses for right child
    Losses($r(g)$) += $N(g) \setminus \hat{N}(r(g))$

climb( c, g )
  // polytomy speciation losses
  if ( $M(c) \notin L(T_S)$ && $M(c) \neq M(g)$ )
    Losses(c) += $C(M(c)) \setminus N(c)$
  SpeciesNode $x \in N(c)$
  if ( $x == M(g) \,||\, p(x) == M(g)$ )
    return n
  while ( $p(x) \neq M(g)$ )
    // skipped losses
    if ( $p(x) \neq M(c)$ )
      Losses(c) += Siblings(x)
    // climb
    $x = p(x)$
  return $\{x\}$
```

Theorem 5.1. *Alg. 5.1 computes required and conditional duplications in $O(|V_G| \cdot (k_S + h_S))$, where k_S is the outdegree of the largest polytomy in T_S, and h_S is the height of T_S.*

Proof. At every internal node $g \in V_G$, $N(g)$ is initialized with $\hat{N}(l(g)) \cup \hat{N}(r(g))$. $|\hat{N}(\cdot)|$ is bounded by k_S. Using a suitable data structure, this step can be achieved in $O(\log(k_S))$ time per node. The *climb* routine is applied to every node in T_G. For any given path from $l \in L(T_G)$ to $r = root(T_G)$, we will climb in total from $M(l)$ to $M(r)$. Thus the total cost of calls to *climb* is $O(|V_G| \cdot h_S)$.

Using fast Least Common Ancestor queries, $M(\cdot)$ can be calculated in $O(|V_G|)$ time for the entire tree.[3] Once $M(\cdot)$ has been calculated, testing for conditional duplications takes constant time per node. Testing for required duplication requires calculation of the intersection of $\hat{N}(l(g))$ and $\hat{N}(r(g))$. This operation takes $O(k_S)$ per node. Combining these, the total running time is $O(|V_G| \cdot (k_S + h_S))$. □

5.2. Minimizing Combined Losses

In the previous section, we presented a low time complexity algorithm to infer an optimal assignment of explicit losses. Although there may be more than one minimum cost assignment, Alg. 5.1 only finds one since it obtains a minimum cost assignment by placing losses as close to $root(T_G)$ as possible.

Next we present an algorithm that considers all possible loss assignments to obtain the minimum number of combined losses, which is always less than or equal to the minimum number of explicit losses. Unlike Alg. 5.1, this algorithm can find all possible optimal assignments, but with increased computational complexity: the worst case running time is exponential in k_S. However, the implementation is fast in practise because k_S is typically small and pathological cases rarely occur. Further speedups can be obtained by memoization.

The basic approach to minimizing combined losses is illustrated using Fig. 6(c). Only losses associated with polytomies, that is losses inferred by rules 3 and 4 from Section 5.1, can be associated with more than one edge in the tree. Consider the polytomy losses $lost\_B$, $lost\_C$ and $lost\_D$ in the right subtree. Because B, C and D are siblings in Fig. 6(a), they can be combined if they can be placed on the same edge in the gene tree. Note that $lost\_B$ can be moved below node 3, resulting in two copies of $lost\_B$, one in each subtree of 3. One copy

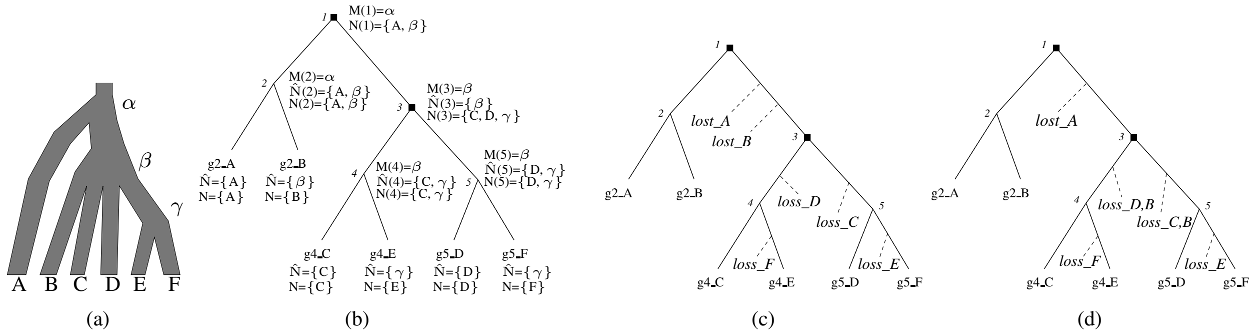


Fig. 6. **(a)** A species tree with a polytomy, β. **(b)** A hypothetical gene tree that has been reconciled with the species tree in (a). This gene tree is annotated with the mappings $M(\cdot)$, $\hat{\mathrm{N}}(\cdot)$ and $N(\cdot)$. Losses are not represented in this tree. **(c)** The gene tree (b) annotated with one possible optimal placement of explicit losses. Losses in the left subtree are not displayed. **(d)** The gene tree (b) annotated with one possible optimal placement of combined polytomy losses. Losses in the left subtree are not displayed.

of $lost\_B$ can be combined with $lost\_C$, while the other can be combined with $lost\_D$. This new placement of losses reduces the number of inferred losses from six to five, as shown in Fig. 6(d). These combined losses are represented by $lost\_D, B$ and $lost\_C, B$.

Each polytomy loss is associated with a particular *Polytomy Connected Component (PCC)* in the gene tree, defined as follows: A node $v \in V_G$ is the *top node* of a distinct PCC if $M(v)$ is a polytomy and $(v = root(T_G) \vee M(v) \neq M(p(v)))$. Let X be the set of nodes that contains v and all $g \in V_G$ such that $g <_G v$ and $M(p(g)) = M(v)$. Then $Y = \{(x, (p(x))| x \in X \setminus root(T_G)\}$ is the PCC with top node v. Note that Y is a contiguous set of edges in T_G. If a loss is inferred at some edge $e \in Y$, it can be placed on e or on any edge below e in Y. For example, the gene tree in Fig. 6(b), has one PCC. Its top node is node 3, and it contains the nodes $\{3, 4, 5, g4\_C, g4\_E, g5\_D, g5\_F\}$.

Alg. 5.2 uses a dynamic program to find the optimal placement for all polytomy losses in a PCC such that they can be combined to minimize the number of losses in the gene tree. To obtain the minimum number of combined losses, we first use a modification of Alg. 5.1 to calculate $\hat{N}(\cdot)$ and $N(\cdot)$, and to infer binary losses with rules 1 and 2 (but not 3 and 4) from Section 5.1. Combined losses are then inferred by calling Alg. 5.2 on the top node of each PCC in the reconciled gene tree.

In Alg. 5.2, a postorder traversal of a PCC is used to calculate minimum cost for any combination of species which could be lost at or below a given node. This cost is stored in a global variable Γ. The loss assignment associated with each cost entry in Γ is stored in Υ. For a node at the bottom of a PCC, the cost is 1 if one or more species must be lost, and 0 if no species are lost. The cost at an internal node g is the sum of the costs for its children, plus the potential cost of a loss at g.

After calculating Γ and Υ, an optimal loss assignment is selected using a preorder traversal of the PCC. Alg. 5.2 calculates a single optimal loss assignment for one PCC. The general algorithm returns all optimal assignments. The worst case running time is $O(|V_G| 2^{3k_S})$. The exponential term is due to the enumeration of the power set of $C(\mu) \setminus \tilde{N}(g)$ in *ProcessComponent* followed by the nested enumeration of two additional power sets in *CalculateCost*. The running time on real data sets is reasonable because these sets are typically small and enumerating the smallest elements of the power set first allows reuse of intermediate results.

Algorithm 5.2.

```
CombinedLosses( g )
    Global component_root = $g$
    // compute costs
    ProcessComponent( $g$, $M(g)$ )
    // add losses using costs
    AddCombinedLosses( $g$, $C(M(g)) \setminus N(g)$ )

$\tilde{N}(g)$
  if ( $g = component\_root$ ) return $N(g)$
  else return $\hat{N}(g)$

ProcessComponent( $g$, $\mu$ )
  if ( $g = component\_root$ || $M(g) = M(p(g))$ )
    ProcessComponent( $l(g)$, $\mu$ )
    ProcessComponent( $r(g)$, $\mu$ )
  else
    $component\_leaf = true$
  foreach $f \in Pow(C(\mu) \setminus \tilde{N}(g))$
    if ( $component\_leaf$ && $f = \emptyset$ )
      $\Gamma(g, f) = 0$
    else if ( $component\_leaf$ )
      $\Gamma(g, f) = 1$
  else
    $\Gamma(g, f)$ = CalculateCost(g, f)

CalculateCost( g, $f_{\text{in}}$ )
 $k^* = \infty$
 if ( g is a Required Duplication )
   foreach $x \in Pow(f_{\text{in}})$
     $k = \sum_{c \in C(g)}($ $\Gamma($ c, $(x \cup \tilde{N}(g)) \setminus \tilde{N}(c)$ ) )
          + $(1|f_{\text{in}} \not\equiv x))$
     if ( $k < k^*$ )
       $k^* = k$; $f^* = x$
   $\Upsilon_l(g, f_{\text{in}}) = (f^* \cup \tilde{N}(g)) \setminus \tilde{N}(l(g))$
   $\Upsilon_r(g, f_{\text{in}}) = (f^* \cup \tilde{N}(g)) \setminus \tilde{N}(r(g))$
 else
   foreach $x \in Pow(f_{\text{in}})$
     foreach $f_{\text{left}} \in Pow(x)$
       $k = \Gamma($ $l(g)$, $f_{\text{left}}$ )
            + $\Gamma($ $r(g)$, $x \setminus f_{\text{left}}$ )
            + $(1|f_{\text{in}} \not\equiv x))$
       if ( $k < k^*$ )
         $k^* = k$; $f^* = x$; ${f_{\text{left}}}^* = f_{\text{left}}$
   $\Upsilon_l(g, f_{\text{in}}) = {f_{\text{left}}}^*$
   $\Upsilon_r(g, f_{\text{in}}) = (f^* \setminus {f_{\text{left}}}^*)$
 return $k^*$

AddCombinedLosses( g, $f_{\text{in}}$ )
 // lose genes at the edge above g
 if ( $g = component\_root \,||\, M(g) = M(p(g))$ )
   Losses(g) += $f_{\text{in}} \setminus (\Upsilon_l(g, f_{\text{in}}) \cup \Upsilon_r(g, f_{\text{in}}))$
   AddCombinedLosses( $l(g)$, $\Upsilon_l(g, f_{\text{in}})$ )
   AddCombinedLosses( $r(g)$, $\Upsilon_r(g, f_{\text{in}})$ )
 else
   Losses(g) += $f_{\text{in}} \setminus \hat{N}(g)$
```

5.3. Related work

To our knowledge, the results above are the first formal algorithms for reconciliation with non-binary species trees. Our approach is similar in flavor to a recent algorithm to root and correct an unrooted gene tree, given a rooted species tree,[4] in that both algorithms use set-based mappings. We propose two such mappings, N and $\hat{N}$, of size bounded above by k_S. The M-mapping in Ref. 4 is equivalent to N. There is no equivalent to $\hat{N}$. Instead, they use a set Z that is $O(V_S)$ and resembles the naïve solution proposed and rejected in Section 4. Although the mappings in both papers are similar, the goals and the algorithmic results differ.

6. EMPIRICAL RESULTS

We have implemented the algorithms described above in a new version of our software tool, NOTUNG, and tested it on several data sets.

First, we confirmed that our non-binary algorithms perform identically to Least Common Ancestor reconciliation when applied to binary trees. We tested the new algorithms in NOTUNG on a benchmark of 15 well-studied, binary trees[5, 17, 32, 38] and verified that the results were the same as those generated by the binary version of NOTUNG, as well as those of the original authors.

Second, we considered performance. The worst case running time of Alg. 5.2 is exponential in k_S. The performance of our preliminary implementation is reasonable for species trees with $k_S \leq 12$. We tested NOTUNG on full trees in TreeFam 3.0[19] with a species tree obtained from the NCBI taxonomy.[46] The time required to reconcile the 1173 gene trees that correspond to species trees with $k_S \leq 12$ was 2'07" for the explicit loss model and 5'21" for the combined loss model on a 3.2ghz OptiPlex GX620 computer. One tree from this data set corresponded to a species tree with a polytomy of size 15 and was not included.

Finally, we reconciled binary gene trees with three species trees that contain well-studied polytomies.[18, 34, 45] All three studies applied statistical tests to verify that the species tree of interest contained a true polytomy. We constructed gene trees for sequence families drawn from each species data set and reconciled them with the non-binary species trees, which were transcribed directly from the source articles. Table 1 shows the number of leaves (l) in each tree, the size of the maximum polytomy (k_S) in each species tree, the number of duplications obtained by binary reconciliation (B), the number of required duplications predicted by our algorithm (R), and the number of combined losses. The globin tree has three equally parsimonious loss assignments. All others have one. As predicted, binary reconciliation substantially overestimates required duplications.

Table 1. Empirical Results

Gene family	Tree		Dupl.s		Losses
	l	k_S	B	R	
Neoaves[34]	12	10	-	-	-
cytochrome B	9	-	4	0	0
globin	17	-	7	4	7
Auklets[45]	5	4	-	-	-
cox1	5	-	1	0	0
cytochrome B	5	-	2	0	0
NADH-6	5	-	2	0	0
Anolis[18]	50	6	-	-	-
NADH-2	50	-	13	7	17

7. DISCUSSION

In this work, we have presented novel algorithms for the reconciliation of binary gene trees with non-binary species trees founded on current theories of deep coalescence and incomplete lineage sorting.[16,23,31,43,44] Our algorithms are both space and time efficient. They have been implemented in a new version of our software tool, NOTUNG. To our knowledge, these are the first formal algorithms for non-binary trees.

Our algorithms are of immediate use to researchers using phylogenetic analysis in a broad range of biological endeavors and are promising for further algorithmic development. Our definitions of required and conditional duplications and the mappings N and $\hat{N}$ provide a foundation for probabilistic models of non-binary reconciliation. Such models would complement the parsimony framework presented here. Probabilistic approaches,[1,2] which assume homogeneous rates, are appropriate for data sets in which duplication and loss are neutral, stochastic processes. Parsimony is better suited to data sets in which duplication and loss are rare due to selective pressure. A probabilistic framework provides a natural setting for incorporating sequence data directly into the reconciliation process, but has the disadvantage that it is both computation and data intensive. A complete phylogenetic toolkit should include both approaches.

Several other problems remain for future work. Our approach assumes that all binary resolutions of a polytomy are equally likely. A more general approach would include models that deviate from a uniform distribution. In addition, non-binary tree models that include horizontal gene transfer as well as gene duplication and loss are needed. Finally, reconciliation of non-binary gene trees with (1) binary and (2) non-binary species trees should also be investigated. Solutions to the former have been suggested;[4,6,10] our solution[10] also has been implemented in NOTUNG. Berglund-Sonnhammer *et al.*[4] proposed a particular formulation of the latter problem and showed that it leads to an NP-complete subproblem. The hardness of the general problem remains open and formal algorithms (or approximation algorithms) are needed.

With the availability of sequences from many closely related genomes, it is increasingly apparent that the histories of individual genes differ and that discordance between gene and species trees is common. Software tools that are sufficiently flexible to handle this situation are needed.[35] The work presented here offers this flexibility.

ACKNOWLEDGMENTS

We thank S. Hartmann, T. Hladish, R. Hoberman, H. B. Nicholas, Jr., M. Sanderson, T. Vision, R. Schwartz, and S. Sridhar for helpful discussions, and R. Cintron, H. B. Nicholas, Jr., and J. Nam for providing phylogenetic trees for the experimental analysis. This work was supported by NIH grant 1 K22 HG 02451-01 and a David and Lucille Packard Foundation fellowship.

References

1. L. Arvestad, A. Berglund, J. Lagergren, and B. Sennblad. Bayesian gene/species tree reconciliation and orthology analysis using MCMC. *Bioinformatics*, 19 Suppl 1:i7–15, 2003.
2. L. Arvestad, A. Berglund, J. Lagergren, and B. Sennblad. Gene tree reconstruction and orthology analysis based on an integrated model for duplications and sequence evolution. In *RECOMB*, pages 326–335, 2004.
3. M. Bender and M. Farach-Colton. Least common ancestors revisited. In *Latin '00*, pages 88–94, 2000.
4. A. Berglund, P. Steffansson, M. Betts, and D. Liberles. Optimal gene trees from sequences and species trees using a soft interpretation of parsimony. *J Mol Evol*, 63(2):240–250, Aug 2006.
5. R. Bourgon, M. Delorenzi, T. Sargeant, A. Hodder, B. Crabb, and T. Speed. The serine repeat antigen SERA gene family phylogeny in *Plasmodium*: the impact of GC content and reconciliation of gene and species trees. *Mol Biol Evol*, 21(11):2161–2171, Nov 2004.
6. W. Chang. Reconciling gene tree with apparent polytomies. Master's thesis, Iowa State University, Ames, IA, 2005.
7. K. Chen, D. Durand, and M. Farach-Colton. Notung: A program for dating gene duplications and optimizing gene family trees. *J Comput Biol*, 7(3/4):429–447, 2000.
8. J. P. Demuth, T. De Bie, J. E. Stajich, N. Cristianini, and M. W. Hahn. The evolution of mammalian gene families. *PLoS ONE*, 1:e85, 2006.
9. J. Dufayard, L. Duret, S. Penel, M. Gouy, F. Rechenmann, and G. Perriere. Tree pattern matching in phylogenetic trees: automatic search for orthologs or paralogs in homologous gene sequence databases. *Bioinformatics*, 21(11):2596–2603, Jun 2005.
10. D. Durand, B. Halldorsson, and B. Vernot. A hybrid micro-macroevolutionary approach to gene tree reconstruction. *J Comput Biol*, 13(2):320–335, 2006. A preliminary version appeared in RECOMB 2005, LNBI 3500, Springer Verlag, 250–264.
11. O. Eulenstein, B. Mirkin, and M. Vingron. Duplication-based measures of difference between gene and species trees. *J Comput Biol*, 5:135–148, 1998.
12. M. Goodman, J. Czelusniak, G. Moore, A. Romero-Herrera, and G. Matsuda. Fitting the gene lineage into its species lineage, a parsimony strategy illustrated by

cladograms constructed from globin sequences. *Syst Zool*, 28:132–163, 1979.

13. R. Guigo, I. Muchnik, and T. Smith. Reconstruction of ancient phylogenies. *Mol Phylogenet Evol*, 6:189–213, 1996.
14. M. Hallett and J. Lagergren. New algorithms for the duplication-loss model. In *RECOMB*, pages 138–146, 2000.
15. G. Hoelzer and D. Melnick. Patterns of speciation and limits to phylogenetics resolution. *Trend Ecol Evol*, 9(33):104–107, 1994.
16. R. Hudson. Gene genealogies and the coalescent process. In *Oxford surveys in evolutionary biology*, volume 7, pages 1–44. Oxford University Press, 1990.
17. A. Hughes. Phylogenetic tests of the hypothesis of block duplication of homologous genes on human chromosomes 6, 9, and 1. *Mol Biol Evol*, 15(7):854–70, 1998.
18. T. Jackman, A. Larson, K. De Queiroz, and J. Losos. Phylogenetic relationships and tempo of early diversification in *Anolis* lizards. *Syst. Biol.*, 48(2):254–285, 1999.
19. H. Li et al. TreeFam: a curated database of phylogenetic trees of animal gene families. *Nucleic Acids Res*, 34(Database issue):572–580, Jan 2006.
20. J. Lyons-Weiler and M. Milinkovitch. A phylogenetic approach to the problem of differential lineage sorting. *Mol Biol Evol*, 14(9):968–975, 1997.
21. B. Ma, M. Li, and L. Zhang. From gene trees to species trees. *SIAM J. on Comput.*, 2000.
22. W. Maddison. Reconstructing character evolution on polytomous cladograms. *Cladistics*, 5:365–377, 1989.
23. W. Maddison. Gene trees in species trees. *Syst. Biol.*, 46(3):523–536, 1997.
24. D. Melnick, G. Hoelzer, R. Absher, and M. Ashley. mtDNA diversity in Rhesus monkeys reveals overestimates of divergence time and paraphyly with neighboring species. *Mol Biol Evol*, 10(2):282–295, Mar 1993.
25. B. Mirkin, I. Muchnik, and T. Smith. A biologically consistent model for comparing molecular phylogenies. *J Comput Biol*, 2:493–507, 1995.
26. J. Nam and N. Masatoshi. Evolutionary change in the numbers of homebox genes in bilateral animals. *Mol Biol Evol*, 22(12):2386–2394, 2005.
27. R. Page. Maps between trees and cladistic analysis of historical associations among genes, organisms and areas. *Syst Zool*, 43:58–77, 1994.
28. R. Page. GeneTree: comparing gene and species phylogenies using reconciled trees. *Bioinf.*, 14(9):819–20, 1998.
29. R. Page and M. Charleston. From gene to organismal phylogeny: Reconciled trees and the gene tree/species tree problem. *Mol Phylogenet Evol*, 7:231–240, 1997.
30. R. Page and E. Holmes. *Molecular Evolution: A phylogenetic approach.* Blackwell Science, 1998.
31. P. Pamilo and M. Nei. Relationships between gene trees and species trees. *Mol Biol Evol*, 5(5):568–583, 1988.
32. M. Pebusque, F. Coulier, D. Birnbaum, and P. Pontarotti. Ancient large-scale genome duplications: phylogenetic and linkage analyses shed light on chordate genome evolution. *Mol Biol Evol*, 15(9):1145–1159, 1998.
33. G. Perrière, L. Duret, and M. Gouy. HOBACGEN: database system for comparative genomics in bacteria. *Genome Research*, 10:379–385, 2000.
34. S. Poe and A. Chubb. Birds in a bush: five genes indicate explosive evolution of avian orders. *Evolution*, 58(2):404–415, 2004.
35. D. Pollard, V. Iyer, A. Moses, and M. Eisen. Widespread discordance of gene trees with species tree in *Drosophila*: evidence for incomplete lineage sorting. *PLoS Genet*, 2(10):e173, Oct 2006.
36. F. Ronquist. Parsimony analysis of coevolving species associations. In Roderic D. M. Page, editor, *Tangled Trees: Phylogeny, Cospeciation and Coevolution*, chapter 2, pages 22–64. Univ of Chicago Press, 2002.
37. C. Roth, M. Betts, P. Steffansson, G. Sælensminde, and D. Liberles. The adaptive evolution database (TAED): a phylogeny based tool for comparative genomics. *Nucleic Acids Res*, 33:D495–D497, 2005.
38. I. Ruvinsky and L. Silver. Newly indentified paralogous groups on mouse chromosomes 5 and 11 reveal the age of a T-box cluster duplication. *Genomics*, 40:262–266, 1997.
39. W. Salzburger, A. Meyer, S. Baric, E. Verheyen, and C. Sturmbauer. Phylogeny of the Lake Tanganyika cichlid species flock and its relationship to the Central and East African haplochromine cichlid fish faunas. *Syst Biol*, 51(1):113–135, Feb 2002.
40. D. Searls. Pharmacophylogenomics: genes, evolution and drug targets. *Nat Rev Drug Discov*, 2(8):613–623, 2003.
41. J. Slowinski and R. Page. How should species phylogenies be inferred from sequence data? *Syst Biol*, 48(4):814–825, Dec 1999.
42. U. Stege. Gene trees and species trees: The gene-duplication problem is fixed-parameter tractable. In *WADS*, LNCS 1663, pages 288–293, 1999.
43. F. Tajima. Evolutionary relationship of dna sequences in finite populations. *Genetics*, 105(2):437–460, Oct 1983.
44. N. Takahata and M. Nei. Gene genealogy and variance of interpopulational nucleotide differences. *Genetics*, 110(2):325–344, Jun 1985.
45. H. Walsh, M. Kidd, T. Moum, and V. Friesen. Polytomies and the power of phylogenetic inference. *Evolution*, 53(3):932–937, 1999.
46. D. Wheeler et al. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, 33(Database issue):D39–45, Jan 2005.
47. D. Wheeler, R. Hope, S. Cooper, G. Dolman, G. Webb, C. Bottema, A. Gooley, M. Goodman, and R. Holland. An orphaned mammalian beta-globin gene of ancient evolutionary origin. *Proc Natl Acad Sci U S A*, 98(3):1101–1106, Jan 2001.
48. L. Zhang. On a Mirkin–Muchnik-Smith conjecture for comparing molecular phylogenies. *J Comput Biol*, 4:177–188, 1997.
49. C. Zmasek and S. Eddy. ATV: display and manipulation of annotated phylogenetic trees. *Bioinformatics*, 17(4):383–4, Apr 2001.
50. C. Zmasek and S. Eddy. A simple algorithm to infer gene duplication and speciation events on a gene tree. *Bioinformatics*, 17(9):821–8, Sep 2001.